**Automating Application Deployments to Various Environments**

* Factors to consider:
  + Code Changes from Developer
  + How to create a software package
  + How is infrastructure organized for your project
  + Repeatable Deployments

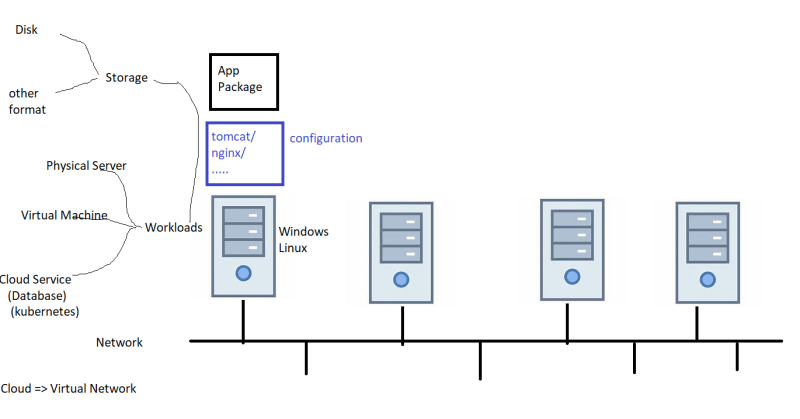
**Code Changes from Developer**

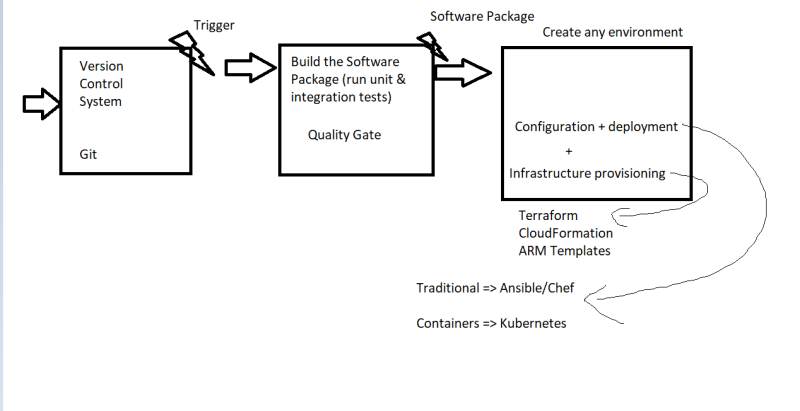
* For every Code Change at a bare minimum, we need to do Continuous Integration.
* Periodically (once in a day or twice in a day or even more) we can combine the changes done at create the infrastructure for various QA environments and run the automated system tests
* Where does developer Submit the Code Changes
  + Git

**How to Create a software Package**

* This is the step which depends on the Programming Language/Technology which your application is developed on
* From the CI/CD Perspective it is all about running one command after installing necessary software’s
* We need to ensure to also run Static Code Analysis (Information, Warning, Critical)
* We need to publish Code Coverage Reports (percentages)
* Quality Gate => Failing the build when it doesnot satisfy Code Analysis & Coverage numbers
* Tools
  + Maven, Gradle => Java
  + Dotnet => Msbuild, dotnet
  + Unittest:
    - java => junit
    - dotnet => nunit, mstest ….
    - python => pytest
  + SonarQube => Static Code Analysis & Code Coverage reports

**Infrastructure Provisioning**



Commonly used infra  
  


**Day To Day Activities**

* For every project we have
  + Day builds => CI for sure (for every develepor commit/ every hour)
  + Night Build => Continuous Delivery => System Test Environment, Peformance Test
  + Weekly Builds => Load/stress tests
* DevOps Engineer:
  + You will me managing atleast 3-5 different projects CI/CD pipelines (50-60%)
  + Ongoing DevOps Activites (infraprovisiong, configuration management, monitoring etc) => (30-40 %)
* Cloud Admin:
  + Creating Infrastructure
  + Enabling Backups
  + Enabling High Availability
  + Network Configurations etc
  + Managing Application in Live Environments